

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-236614

(43) Date of publication of application : 23.08.2002

(51)Int.Cl. G06F 12/08  
G06F 17/16

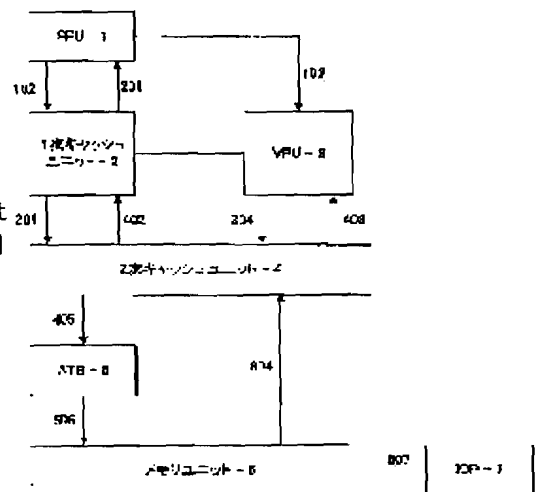
(21)Application number : 2001-033857 (71)Applicant : NEC CORP  
(22)Date of filing : 09.02.2001 (72)Inventor : KOYANAGI HISAO

## (54) CACHE CONTROL METHOD AND CACHE CONTROL CIRCUIT

(57)Abstract:

**PROBLEM TO BE SOLVED:** To provide a cache control method and a cache control circuit that can solve a conventional technical problem and acquire a performance improving effect by a cache memory even with a program which does not guarantee locality of reference in a strict sense.

**SOLUTION:** This cache control circuit of an information processor with the cache memory is provided with a first register for holding the value that specifies a vector load command access range; a second register for holding vector length; a third register for holding distance; a circuit for carrying out determination of registration or nonregistration of vector load command data to a cache on the basis of the values of the first, second and third registers; a fourth register for holding the determined result; and an erroneous reply control circuit for selectively performing the control of registration or nonregistration to the cache according to the determined result of the fourth register.



## LEGAL STATUS

[Date of request for examination] 17.01.2002

[Date of sending the examiner's decision of rejection] 08.06.2005

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 3770091

[Date of registration] 17.02.2006

[Number of appeal against examiner's decision of rejection] 2005-013125

[Date of requesting appeal against examiner's decision of rejection] 08.07.2005

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号  
特開2002-236614  
(P2002-236614A)

(43) 公開日 平成14年8月23日 (2002.8.23)

(51) Int.Cl. <sup>7</sup>	識別記号	F I	テームト* (参考)
G 0 6 F 12/08	5 1 7	G 0 6 F 12/08	5 1 7 B 5 B 0 0 5
	5 0 1		5 0 1 D 5 B 0 5 6
	5 5 9		5 5 9 B
17/16		17/16	C

審査請求 有 請求項の数 3 O L (全 12 頁)

(21) 出願番号 特願2001-33857(P2001-33857)

(22) 出願日 平成13年2月9日 (2001.2.9)

(71) 出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72) 発明者 小柳 尚夫

東京都港区芝五丁目7番1号 日本電気株式会社内

(74) 代理人 100079005

弁理士 宇高 克己

Fターム(参考) 5B005 JJ13 KK12 MM01 QQ00

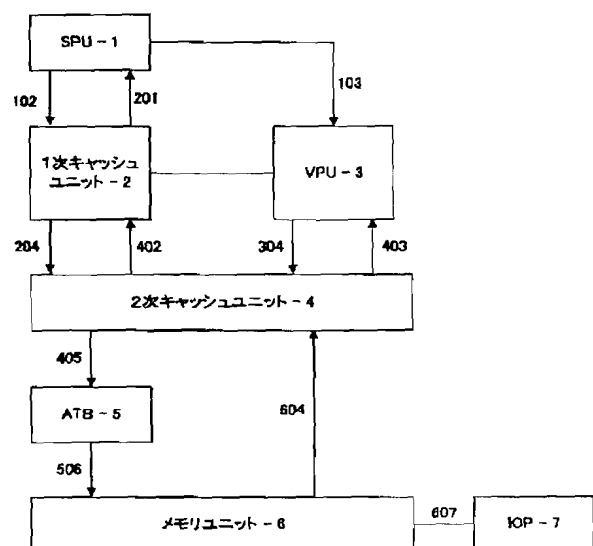
5B056 BB32 FF07 FF10 HH00

(54) 【発明の名称】 キャッシュ制御方法及びキャッシュ制御回路

(57) 【要約】

【課題】 従来技術の問題点を解消し、参照の局所性が厳密な意味で保証されないプログラムであっても、キャッシュメモリによる性能向上効果を得ることができるキャッシュ制御方法及びキャッシュ制御回路を提供すること。

【解決手段】 キャッシュメモリを有する情報処理装置のキャッシュ制御回路であって、ベクトルロード命令アクセス範囲を規定する値を保持する第1のレジスタと、ベクトル長を保持する第2のレジスタと、ディスタンスを保持する第3のレジスタと、前記第1、第2及び第3のレジスタの値に基づいてベクトルロード命令のデータをキャッシュに対して登録または非登録の判定を実行する回路と、該判定結果を保持する第4のレジスタと、該第4のレジスタの判定結果にしたがって、キャッシュへの登録または非登録の制御を選択的に実行するミスプライ制御回路と、を具備する。



## 【特許請求の範囲】

【請求項1】 キャッシュメモリを有する情報処理装置のキャッシュ制御方法において、ベクトルロード命令アクセス範囲を規定する値、ベクトル長の値、そしてディスタンス値、の各々の値を基礎として、ベクトルロード命令のデータをキャッシュへ登録するかまたは非登録とするかの判定を行い、その判定結果に従ってキャッシュへの登録または非登録の制御を選択的に実行することを特徴とするキャッシュ制御方法。

【請求項2】 キャッシュメモリを有する情報処理装置のキャッシュ制御回路において、ベクトルロード命令アクセス範囲を規定する値を保持する第1のレジスタと、ベクトル長を保持する第2のレジスタと、ディスタンスを保持する第3のレジスタと、前記第1、第2及び第3のレジスタの値に基づいてベクトルロード命令のデータをキャッシュに対して登録または非登録の判定を行う回路と、該判定結果を保持する第4のレジスタと、該第4のレジスタの判定結果に従って、キャッシュへの登録または非登録の制御を選択的に実行するミスリプライ制御回路と、を具備することを特徴とするキャッシュ制御回路。

【請求項3】 キャッシュメモリを有する情報処理装置のキャッシュ制御回路において、キャッシュアクセスの制限使用回数を格納する第5のレジスタと、キャッシュの各エントリに対応してキャッシュアクセス回数の計測結果を保持する第6のレジスタ群と、その値を更新する制御回路と、アクセスしたキャッシュエントリに対応する第6のレジスタ値と第5のレジスタ値とを比較した結果から、キャッシュへの登録または非登録の判定を行う制御回路と、その結果を保持する第4のレジスタと、該第4のレジスタの判定結果にしたがって、キャッシュへの登録または非登録の制御を選択的に実行するミスリプライ制御回路と、を具備することを特徴とするキャッシュ制御回路。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、各種情報処理システムにおいて使用されるキャッシュメモリを制御するためのキャッシュ制御方法及びキャッシュ制御回路に関する。

## 【0002】

【従来の技術】各種情報処理システムにおけるメモリアccessのボトルネック解消のために、キャッシュと呼ばれる小容量で高速のバッファメモリを利用する技術は、古くからほとんどのシステムで使用されている。キャッシュメモリを使用する効果は、以下の2つである。その第1は、主メモリ等に比べて小容積に構成し得ることから、プロセッサの近く、例えば、同一LSI内部に配置できることになり、オペランドデータのレイテンシー（Latency）の大幅短縮が実現できることであ

る。

【0003】第2の効果は、特に、ストアイン型キャッシュのようにメモリへの書き込みを常時には行わない場合、メモリアccessに必要なスループットを抑えることができる。したがって、メモリ構造を簡素化でき、コスト面で有利であることと、逆にいえば、メモリアccess頻度を抑えることで、メモリアccessのレイテンシーの競合を回避できるため、性能向上が実現できることである。

【0004】このキャッシュメモリ方式が有効であるためには、以下に示す実際のプログラムにおけるメモリアccessの振る舞いが、通常は一定の性質を持つものであるとの前提が必要である。それは、参照の局所性（以下、ローカリティともいう）というメモリアccessの性質によるものである。すなわち、空間的ローカリティ（参照されるアドレスはある部分に固まっていること）と、時間的ローカリティ（参照されるアドレスは、ある時間内に集中していること）である。

【0005】ここでキャッシュミス時の動作について簡単に説明する。ただし、キャッシュメモリの動作については、多くの文献があり、例えば“並列コンピュータ”

天野英晴著（昭晃堂）に詳しく説明されているので、本発明に直接関係しない動きに関しては説明を割愛する。

【0006】まず、既にキャッシュに登録されている同一キャッシュラインアドレスのラインは追い出される。ストアスルー型キャッシュの場合は、メモリとの一貫性が常時保たれているため、新しいラインをキャッシュに上書きするだけで済む。しかし、ストアイン型キャッシュの場合には、メモリとの一貫性が保たれていないので、追い出されるキャッシュラインをメモリに書き込む必要がある。

【0007】また、新しいラインのリフィルにラインデータ分のブロックロードリクエストが発生する。メモリに対して、無駄なデータを持ってくる処理が入るが、その一方で暗示的なプリフェッチとなるという長所もあるため、その得失は一概には断じられない。

【0008】この動作に関しては図5において説明しており、“A”、“B”、“W”は、ロード（以下、LDともいう）命令が、各々メモリアccessリクエストを発行するステージ、キャッシュアクセスを終わるステージ、レジスタへの書き込みステージに存在することを意味する。

【0009】まず、LD-1はキャッシュヒット（hit）時のタイミングであり、この場合、わずかに3サイクルで終了する。LD-2は、キャッシュミス（miss）した場合の動作であり、メモリに対してリフィルデータを要求し、ターゲットとなるデータが最初に帰ってくる場合が、一般的であり、その時点で、LD-2はWステージに移行することができる。しかし、リフィルの

ために、1ライン分のデータをキャッシュに書き込むための時間が必要であり、その間は、後続のキャッシュアクセス命令の発行が不可となる。LD-3は、キャッシュにデータを登録しないキャッシュバイパスロードのタイミングであり、LD-2との違いは、ミスした場合に、ミスプライデータをキャッシュに登録しない点にある。

【0010】参照の局所性がないメモリアクセスの場合、リフィルデータのキャッシュ書き込み時における後続キャッシュアクセス命令の発行が抑止されることがない点から、このケースだけを見ればLD-3が性能上優れていると言える。

【0011】以上のようなキャッシュ制御は、前提としている参照の局所性が崩れると、以下のような問題点が発生する。問題点1) 1回しかLDしないラインをキャッシュに入れることは、キャッシュにデータを登録しない場合に比べて、キャッシュラインデータのストアとロードが余計に必要となる。問題点2) まだ有用なキャッシュラインにもかかわらずリブレースしてしまう。

【0012】このような参照の局所性がないといわれているものには、科学技術計算プログラムがある。しかし、近年、マイクロプロセッサにおけるキャッシュ容量は、LSI技術の絶え間ない進歩によって、KB単位からMB単位に増強されている。また、キャッシュを考慮したプログラムの最適化技術も、このようなキャッシュ容量のトレンドに追従し、ストアイン型キャッシュのマイクロプロセッサにおいても、科学技術計算プログラムの性能が向上してきた。しかし、コンパイラの最適化技術はいくら進んでも、実行前のプログラムからメモリアクセスの振る舞いを認識して、高いキャッシュヒット率を実現するには限界がある。

【0013】ところで、科学技術計算プログラムを効率よく実行する装置として、ベクトル型計算機がある。ベクトル型計算機では、ベクトルレジスタという複数ワードデータを格納するレジスタに対する処理を、ベクトル命令という命令1つで規定できる。例えば、ベクトルロード（以下、VLDともいう）命令は、ベクトル長レジスタ（以下、VLともいう）の値分のロードを実行し、その読み出しデータをベクトルレジスタという複数ワードを集めたレジスタに格納する。逆に、ベクトルストア（以下、VSTともいう）はベクトルレジスタにある複数のワードデータを、メモリに格納する。

【0014】このベクトル型計算機において、例えば〔ディスタンス (distance)〕\*〔VL〕の大きなVLD命令が発行され、それが悉くキャッシュミスする場合、VLD1命令で数多くのキャッシュラインがリブレースされてしまう。その中で、使ったばかりのラインが存在していた場合、時間的局所性を期待してキャッシュに登録されているデータがキャンセルされ、再度度キャッシュラインデータがリフィルされてしまうという、性

能上悲劇的状况に至る。

【0015】

【発明が解決しようとする課題】本発明は、上述のような従来技術の問題点を解消し、参照の局所性が厳密な意味で保証されないプログラムであっても、キャッシュメモリによる性能向上効果を得ることができるキャッシュ制御回路を提供することである。

【0016】

【課題を解決する為の手段】本発明の課題は、キャッシュメモリを有する情報処理装置のキャッシュ制御方法において、ベクトルロード命令アクセス範囲を規定する値と、ベクトル長の値と、そしてディスタンス値と、の各々の値を基礎として、ベクトルロード命令のデータをキャッシュへ登録するかまたは非登録とするかの判定を行い、その判定結果に従ってキャッシュへの登録または非登録の制御を選択的に実行するキャッシュ制御方法によって解決される。

【0017】さらに、本発明の課題は、キャッシュメモリを有する情報処理装置のキャッシュ制御回路において、ベクトルロード命令アクセス範囲を規定する値を保持する第1のレジスタと、ベクトル長を保持する第2のレジスタと、ディスタンスを保持する第3のレジスタと、前記第1、第2及び第3のレジスタの値から、ベクトルロード命令のデータをキャッシュへの登録または非登録の判定を行う回路と、その結果を保持する第4のレジスタと、該第4のレジスタの判定結果に従ってキャッシュへの登録または非登録の制御を選択的に行うミスプライ制御回路と、を具備するキャッシュ制御回路によって解決される。

【0018】また、本発明の課題は、キャッシュメモリを有する情報処理装置のキャッシュ制御回路において、キャッシュアクセスの制限使用回数を格納する第5のレジスタと、キャッシュの各エントリに対応してキャッシュアクセス回数の計測結果を保持する第6のレジスタ群と、その値を更新する制御回路と、アクセスしたキャッシュエントリに対応する第6のレジスタ値と第5のレジスタ値とを比較した結果から、キャッシュへの登録または非登録の判定を行う制御回路と、その結果を保持する第4のレジスタと、該第4のレジスタの判定結果にしたがって、キャッシュへの登録または非登録の制御を選択的に行うミスプライ制御回路と、を具備するキャッシュ制御回路によって有利に解決される。

【0019】本発明にかかるキャッシュ制御方法及び制御回路によれば、ロードリクエストの性質、あるいは、ヒットしたラインの履歴によって、キャッシュメモリに登録させるロードにするか否かをハードウェア（以下、HWともいう）が動的に判断する。

【0020】また、VLD範囲レジスタとRUカウンタ制限値レジスタの更新によって、判断方法をソフトウェア（以下、SWともいう）が明示的に指定することがで

10

20

30

40

50

きる。それによって、発明の背景で説明したような問題を解決し、参照のローカリティが厳密な意味で保証されないプログラムであっても、キャッシュメモリによる性能向上効果を得ることができる。

#### 【0021】

【発明の実施の形態】以下、添付図を参照しつつ本発明にかかるキャッシュ制御装置の実施の形態を説明する。このキャッシュ制御装置は、図1に示されているような基本構成に適用される。スカラプロセッサユニット（以下、SPUともいう）-1は、全命令の解釈／実行指示、及びスカラ命令の実行を行う。1次キャッシュユニット-2は、スカラメモリアクセス系命令で使用するデータのみを格納するものである。ベクトルプロセッサユニット（以下、VPUともいう）-3は、SPU-1の指示にしたがってベクトル命令の実行を行う。

【0022】2次キャッシュユニット-4は、スカラ系／ベクトル系の両方で使われるデータを格納するものである。アドレス変換バッファ（以下、ATBという）-5は、論理アドレスを物理アドレスに変換するバッファである。また、メモリユニット-6は、メモリとメモリアクセス制御部とを含んでいる。IOP-7は入出力処理を行う。

【0023】スカラLDの場合、SPU-1から信号線102でLDリクエストを発行し、1次キャッシュユニット-2内でミスすると、ラインデータのブロックリクエストとして信号線204を介して2次キャッシュユニット-4にデータを探しにゆく。そこでもミスした場合は、信号線405を介してATB5にメモリリクエストを送り、物理アドレスに変換して信号線506を介してメモリユニット-6に送出される。メモリユニット-6から読み出されたデータは、信号線604を介して2次キャッシュユニット-4に戻ってきてラインデータが格納され、信号線402を介して1次キャッシュユニット-2に戻って、そのラインデータが登録される。その時、最初のLD命令のターゲットデータは、信号線201を介してSPU-1に送られる。

【0024】次に、VLD命令の動きについて説明する前に、VLD命令の仕様について、図6に基づいて説明する。VPU-3内にはベクトルレジスタとも呼ばれる複数の要素データを格納できるレジスタファイルが存在する。VLD命令は、Vxで指定されたベクトルレジスタに、SPU-1から送られてくるRy（ディスタンス）とRz（開始アドレス）で規定されたVLというレジスタで指定される個数のメモリロードを一括して行う命令である。

【0025】ここで本発明の実施例を説明するため、図1のような基本構成を用いる。VLというレジスタは、LVLという命令によって、ソフトウェアで書き換えられるレジスタで、VPU-3内に存在する。VLD命令の動きを図1のブロック図にしたがって説明する。

VLD命令はSPU-1で命令デコードされ、その結果、信号線103を介して実行指示がVPU-3に送られる。VPU-3では、VLD命令が処理されるが、その際、開始アドレス、ディスタンス、VL（ベクトル長）が信号線304を介して2次キャッシュユニット-4に送られる。

【0026】2次キャッシュユニット-4では、ミスした場合、該当するラインに対するブロックリクエストになり、スカラLDと同様にATB-5を通過し、メモリユニット-6から信号線604を介してリプライデータを2次キャッシュユニット-4に送る。2次キャッシュユニット-4が受け取ったリプライデータは、対応するキャッシュラインに格納されると同時に信号線403を介してターゲットデータのみがVPU-3に送られ、ベクトルレジスタに格納される。このような処理がVLで指定された数だけ実行される。しかし、2次キャッシュがヒットした場合は、2次キャッシュの読み出しデータだけが信号線403を介してVPU-3に送られる。

【0027】ここで、このVLD命令を、本発明によるVLD範囲レジスタ、または、RUカウンタ制限値レジスタを用いたキャッシュ判定回路によって、キャッシュミスした場合に必ずしもキャッシュに取り込まないようにする。この場合、2次キャッシュのラインサイズ of ブロックロードリクエストが発行されることはなく、信号線405にVLDの要素毎のリクエストが発行され、信号線604を介して送られてくるリプライデータを素通りさせて、信号線403を介して、VPU-3へ送る。

【0028】まず、VLD範囲レジスタによる、本発明における動作について説明する。従来のベクトルメモリアクセスは、全てキャッシュされないものであったが、前述のようにLSI技術の進歩によるキャッシュ容量の大幅増加と、コンパイラによる最適化技術の進歩によって、科学技術計算でもある程度のキャッシュを使って意味のあるような参照局所性が実現できる。しかし、その中でもどうしても局所性を維持するような最適化ができない部分も時には存在する。その様子を図7に示す。

【0029】その局所性のない部分では、広範囲にわたるVLD命令でベクトルオペランドデータをフェッチして、それを演算し、VST命令でメモリに書き戻すといった処理が行われ、1回VLD命令で読んだオペランドは2回以上読み出されることはほとんどない。このような場合、VLDのメモリデータをキャッシュに取り込むことは、無意味である。

【0030】また、図8では、VLD命令のディスタンスが128BでVLが256のケースで、キャッシュ（ラインサイズ128B）に登録したケースについて説明している。VLD命令の要素データは、ラインデータのある1部にのみ存在するため、VLD命令1つのために、メモリLDデータは8B×256（＝2KB）しか

必要でないにもかかわらず、128B×256(=32KB)ものブロックLDが発生し、無駄なLDリクエストが30KB分も行われることになる。さらに、32KB分のキャッシュは、このVLD命令によって、図7で示しているような参照局所性が維持されている部分で必要とされているキャッシュデータが書きつぶされてしまうという悲劇的状况が生まれる。

【0031】このような状況を回避するための別の手段として、キャッシュメモリにデータを登録しないことを最初から命令で明示的に指定するという方法もある。この方法の利点は、アドレス・ディスタンスといった値の着目だけでは不十分な参照の局所性についても考慮できることで、より効果的なキャッシュ制御が可能となる点である。このキャッシュに登録しないVLD命令が実行された場合には、無条件でキャッシュ登録を行わない。ただし、ロードすべきデータがキャッシュメモリに存在していた場合は、当然、キャッシュメモリを読み出す。

【0032】しかし、キャッシュメモリにデータを登録するか否かをSW(ソフトウェア:コンパイラ)で判断することは極めて困難であり、ある程度のHWによるサポートが必要である。そのため、本発明では、VLD範囲レジスタで、VLDのアクセスする領域の広さを局所性の有無と判断する回路の実装によって、図8に示したような非効率的なキャッシュ動作を回避することができ\*

『V Hit LineHit RUカウンタ 備考

0	—	—	A110	無効なライン
1	0	—	No change	
1	1	0	+1	別のラインでHit
1	1	1	A110	Hitして、使用』

このようにして、使用されないキャッシュラインに関しては、RUカウンタの値が上がってゆき、使用される可能性が低いラインであると認識される。このRUカウンタ値がRUカウンタ制限値レジスタを超えない場合は、リプレース対象から外すことによって、使用される可能性の高いラインがキャッシュに残ることになる。

【0036】これらの動作を実現する本特許の特徴となる2次キャッシュの構成について、図2に基づいて説明する。VLD範囲REG-1(値と有効ビットで構成され、有効ビットが点灯していない時、比較結果はnon-activeである)は、乗算器-4によって計算された[VLレジスタ-2の出力]×[ディスタンスレジスタ-3の出力]と比較され、比較器-5の比較結果信号によって、ミス時にキャッシュに登録するか否かを決定する。命令コードレジスタ(OPC)-17の値から命令デコーダー-7によって、VLD信号(信号線706)及びキャッシュ非登録VLD信号(信号線710)を生成する。

【0037】ANDゲート-6は、VLDが範囲を超えるものであったことを示す信号を生成する。ANDゲ-

＊る。

【0033】次に、RUカウンタ制限値レジスタによる時間的局所性に着目したキャッシュ登録制御について説明する。時間的局所性とは、1度アクセスされたデータは、再びアクセスされやすい傾向があるということである。このようなラインは、他のメモリアクセス時のキャッシュミスによって、キャッシュから追い出されないようにしなければならない。このキャッシュにおける追い出しアルゴリズムに関しては、従来技術にあるように、LRUアルゴリズムがある。

【0034】ところが、それは複数のWayで構成されるキャッシュにおいて有用なものであり、ダイレクトマップ構成であると、同一ラインのアクセスに関しては、無条件に追い出されてしまう。さらに、キャッシュ容量の巨大化の方向では、HWコストの面からダイレクトマップ方式のキャッシュが一般的である。

【0035】また、キャッシュ登録データは少なくとも、キャッシュに存在している限り2回以上そのラインにアクセスしないと、そのまま1ライン分のブロックLDのコストだけ、逆効果となる。そこで、最近特にアクセスが頻発に行われているラインであることを判断するために、RUカウンタという情報を各ラインに設ける。RUカウンタの動作論理は下記のようなものである。

ト-9は、RUカウンタ制限値REG-12(値と有効ビットとで構成され、有効ビットが点灯していない時は、比較結果はNon-activeである)とRUカウンタ-30の値を比較する比較器-8の比較結果信号より、時間的局所性の解析結果の信号が生成される。ORゲート-10は、Non-Cachable(キャッシュ非登録)の判断結果信号を出力し、その結果信号をキャッシュ非登録指示フラグ-11が受ける。

【0038】有効フラグ-13は、Aステージに1次キャッシュからのブロックロード、あるいは、VPUからのベクトルメモリアクセス命令のリクエストが有効であることを示し、2Wayセクター-19は命令コードレジスタ(OPC)-17へ格納する値を選択する。

【0039】OPC-17及びアドレスレジスタ(ADR)-14には、Aステージに存在する命令のコードとアドレスが格納される。ORゲート-15は、1次キャッシュからのブロックロード、あるいは、VPUからのベクトルメモリアクセス命令のリクエストが有効の出力を生成する。

【0040】WEフラグ-16の点灯するケースは、ス

トア命令がwriteヒット時と、キャッシュミスプライのデータを書き込む時である。その2つのタイミングをORゲート20で生成している。前者はキャッシュヒット判定部33から、後者はMissREQ制御部36からそれぞれ送られる。また、その時のアドレスは、1次キャッシュ、VPU、Bステージのアドレスレジスタ26、MissREQ制御部36から送られる、各々、1次キャッシュブロックロードアドレス、ベクトルメモリアクセス命令のアドレス、ストアwrite 10 ヒット時のアドレス、キャッシュミスプライのアドレスを、4Wayセクター18で選択するものである。

【0041】WDR (Write Data Register) - 20は、DA (Data Array) - 32に対する書き込みデータレジスタである。2Wayセクター21は、ストアwriteヒット時のストアデータ、キャッシュミスプライデータを選択する。2Wayセクター23は、1次キャッシュ/VPUからのストアリクエストのデータを選択し、ストアデータレジスタ22に格納する。ストアがwrite 20 ヒットした場合、ストアデータレジスタ24で1サイクル持ち回った後、WDR-20に格納し、そしてDA-32に書き込む。有効フラグ25、アドレスレジスタ26、ストアデータレジスタ24、キャッシュミスフラグ27は、いずれもBステージの情報である。

【0042】キャッシュ有効ビット29は、キャッシュのラインが有効であることを示す。アドレスアレイ (AA) - 31は、対応するラインが如何なるアドレスのデータであるかを示す。データアレイ32は、キャッシュデータを格納する。アドレス比較器28は、A 30 AA-31の出力を読んで、アドレスレジスタ14の値と比較し、キャッシュミスを判定し、その結果をキャッシュミスフラグ27に格納する。

【0043】キャッシュヒット判定部33は、Bステージの制御情報を使用して、RUカウンター30を更新するための制御信号を生成する。DA-32から読んだキャッシュデータと、メモリからのキャッシュミスプライデータとを2Wayセクター34が選択し、RDR (Read Data Register) - 35に格納する。

【0044】次に、図3に基づいて、本実施例が前提としている2次キャッシュ構成について説明する。本実施例では、ラインサイズ128B×256エントリ (容量32KB) のダイレクトマップ方式を前提として説明する。このキャッシュの場合、AA (アドレスアレイ) 及びDA (データアレイ) から構成され、AAに格納されるINDEXアドレスに対応するタグアドレスとリクエストのアドレスが比較され、キャッシュヒットの判定が行われる。メモリアクセスリクエストのアドレスは全体で40bitであり、そのうち上位25ビットはTAG 50

アドレス、下位7ビットはラインアドレス、中間の8ビットはINDEXアドレスというように構成される。

【0045】次に、図2中のRUカウンター30の周辺回路について、図4に基づいて説明する。RUカウンター30は、キャッシュのエントリに存在し、ADR-14の値に応じてデコーダ143とセクタ301で選択され、その出力がCachable決定回路に送られる。

【0046】キャッシュヒット判定部33は、INDEXアドレスのデコーダ331、命令デコーダ332を使って、RUカウンター30に対して、キャッシュヒット時の状況を報告する信号を出力する。デコーダ331は、INDEXアドレスの256ビットのデコード信号を生成する。命令デコーダ332は、信号線3321にキャッシュ入りロード命令デコード信号を、そして信号線3322にキャッシュ入りストア命令デコード信号をそれぞれ生成する。信号線3330の信号は、Bステージに存在する有効なメモリロード命令がヒットしたことを示す。信号線3331の信号は、あるキャッシュエントリで、Bステージに存在する有効なメモリロード命令がヒットしたことを示す。また、ストア命令でのwrite ヒット検出信号を信号線3332に生成し、Bステージの次のタイミングでWEフラグを点灯させ、キャッシュにストアデータを書き込む。

【0047】次に、図9に基づいてVL=2の時のVLD命令がキャッシュヒットした場合の動作について説明する。2次キャッシュをアクセスする命令は、Aステージで命令コード、アドレスが、各々、OPC-17、ADR-14に設定される。Bステージで、1st/2nd リクエストが両方ヒットした場合、次のタイミングは両方ともWステージに移行し、読み出しデータがRDR-35を介してVPUに送られる。

【0048】次に、図10に基づいてVL=2の時のVST命令がキャッシュヒットした場合の動作について説明する。Aステージで命令コード、アドレス、ストアデータが、各々、OPC-17、ADR-14、ストアデータ22に設定される。Bステージで、1st/2nd リクエストが両方ヒットした場合、次のタイミングは両方ともストアデータが、ストアデータ22に格納され、WE-16が点灯して、DA-32に書き込まれる。

【0049】次に、図11に基づいてVL=2の時のVLD命令がキャッシュミスした場合の動作について説明する。Aステージで命令コード、アドレスが、各々、OPC-17、ADR-14に格納され、Bステージでミスが判定される。それと同時にVL-2、distance-3の値とADR-26、さらに、VLD範囲REG-1との比較によって、Cachableであることを判定する。ここでのCachable判定条件は、 $\{VLD \text{ 範囲 } REG-1\} > \{VL\} * \{distance\}$



ce)である。キャッシュミス時のメモリリクエストは、1ライン分データのblockロードとしてATBへ送られ、その後ミスプライ有効信号としてターゲットになるワードデータから順に帰ってくる。そして、タイミング⑧及び⑨でWE-16を点灯させ、キャッシュに登録するとともにRDR-35を介してVPUにリプライデータを送出する。さらに、残りのキャッシュラインデータをキャッシュへ書き込むために、タイミング

(10)以後でも、WE-16が点灯する。この場合、従来技術でも説明したように、後続命令の発行を止める必要がある。

【0050】次に、図12に基づいてVL=2の時のVST命令がキャッシュミスした場合の動作について説明する。Aステージで、命令コード、アドレス、ストアデータが、各々OPC-17、ADR-14、ストアデータ22に設定される。Bステージで、1st/2ndリクエストが両方ミスした場合、次のタイミングで両方ともATBへのリクエストとして送出される。

【0051】次に、図13に基づいてVL=2の時のVLD命令がキャッシュミスし、且つNon-cachableとなった場合の動作について説明する。Aステージで命令コード、アドレスが、各々OPC-17、ADR-14に格納され、Bステージでミスが判定される。それと同時にVL=2、distance=3の値とADR-26、さらに、VLD範囲REG-1との比較によって、Non-cachableであることを判定する。ここでのCachable判定条件は、 $[VLD \text{ 範囲 } REG-1] \leq [VL] * [distance]$ である。キャッシュミス時のメモリリクエストは、要求するワードデータのためのリクエストとしてATBへ送られ、その後ミスプライ有効信号として帰ってくる。しかし、Non-cachableであるため、WE-16は点灯せず、リプライデータを単にVPUへ返すだけである。したがって、その後のリフィル動作が発生しないので、その分後続命令の待ちが発生しない点で図11の動作と異なる。

【0052】次に、図14に基づいて、本特許におけるキャッシュ登録判定にかかわるVLD範囲REG-1、RUカウンタ制限値REG-13への命令による書き込み動作について説明する。上記2つのレジスタは、LCRG命令によってセットされる。その命令のYフィールドで指定されたレジスタ内容にしたがって各々のレジスタに値を格納する。本発明における機能を有効にしたい場合は、各々のVビットに“1”を立てる必要がある。一方、LCRG命令の動きとしては、Aステージに到着する1サイクル前にVPUからの指示によって、VLD範囲REG(V, data)-1、RUカウンタ制限値REG(V, data)-13に設定される。

【0053】図15は、図9～図13におけるVLD命令及びVST命令が発生した場合の動作についてフロー

チャートとしてまとめたものである。図左側に示すフローのように、VLD命令リクエストが発生すると、キャッシュヒットか否かが判定され、ヒットした場合は、キャッシュを読んでVPUに返す(図9に対応するフロー)。

【0054】ヒットしない場合には、次いで、Cachableか否かが判定される。Cachableである場合には、ATBへ1ライン分のブロックLDリクエストを発行し、リプライターゲットデータをVPUに返して、キャッシュにブロックデータを書き込む(図11に対応するフロー)。

【0055】他方、Cachableでない場合は、ATBへ必要なワードのみのリクエストを発行し、そしてリプライターゲットデータをVPUへ返す(図13に対応するフロー)。

【0056】図15右側のようにVST命令リクエストが発生すると、キャッシュヒットか否かが判定される。ヒットした場合には、Cacheにデータの書き込みを行う(図10に対応するフロー)。他方、ヒットしない場合には、ATBへメモリ書き込みのリクエストを発行する(図12に対応するフロー)。

【0057】

【効果】本発明にかかるキャッシュ制御方法並びに制御回路によれば、ロードリクエストの性質、あるいは、ヒットしたラインの履歴によって、キャッシュに登録させるロードにするか否かをHWが動的に判断することになる。

【0058】また、VLD範囲レジスタとRUカウンタ制限値レジスタの更新によって、判断方法をSWが明示的に指定することができる。それによって、発明の背景で説明したような問題点を解決し、参照のローカリティが厳密な意味で保証されないプログラムであっても、キャッシュによる性能向上効果を得ることができる。

【図面の簡単な説明】

【図1】本発明の実施例におけるキャッシュ制御回路の全体構成図である。

【図2】本発明の実施例における2次キャッシュ周辺回路の構成図である。

【図3】本発明の実施例における2次キャッシュ構成図である。

【図4】RUカウンタ制御回路周辺構成図である。

【図5】キャッシュリフィル時の性能低下について説明した図である。

【図6】本発明の実施例におけるVLD命令仕様を示す図である。

【図7】科学技術計算におけるメモリアクセスの局所性について説明した図である。

【図8】局所性のないプログラムにおいてキャッシュが意味をなさない例を説明した図である。

【図9】VL=2の時のVLD命令がキャッシュヒット

した時の動作について説明した図である。

【図10】VL=2の時のVST命令がキャッシュヒットした時の動作について説明した図である。

【図11】VL=2の時のVLD命令がキャッシュミスした時の動作について説明した図である。

【図12】VL=2の時のVST命令がキャッシュミスした時の動作について説明した図である。

【図13】VL=2の時のVLD命令がキャッシュミスし、且つ、Noncachableとなった時の動作について説明した図である。

【図14】キャッシュ登録判定にかかわるレジスタへの命令による書き込み動作について説明した図である。

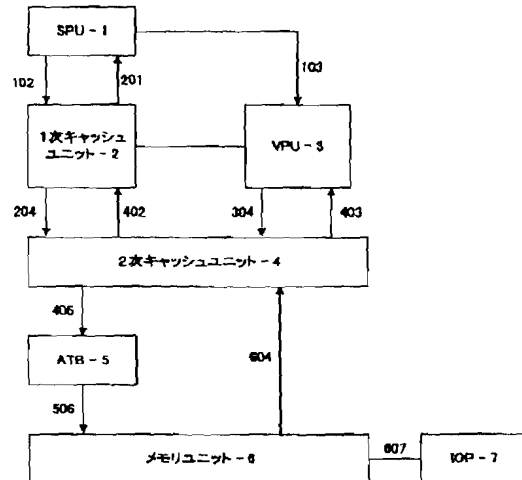
【図15】図9ないし図13に対応するVLD命令及び\*

\* VST命令の動作に関するフロー図である。

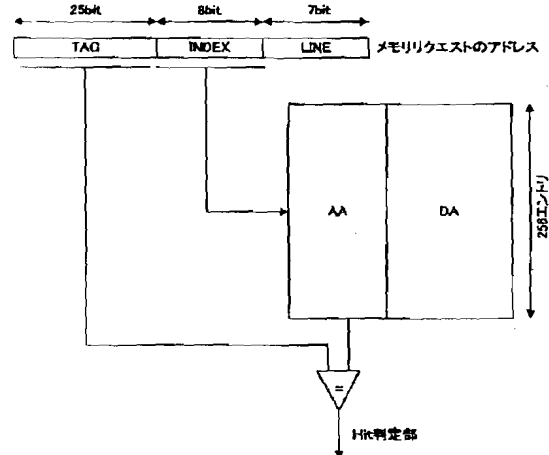
【符号の説明】

SPU スカラブロッサユニット  
VPU ベクトルブロッサユニット  
ATB アドレス変換バッファ  
IOP 入出力処理部  
VLD ベクトルロード  
VL ベクトル長  
AA アドレスアレイ  
DA データアレイ  
WDR 書き込みデータレジスタ  
RDR 読み出しデータレジスタ  
OPC 命令コードレジスタ

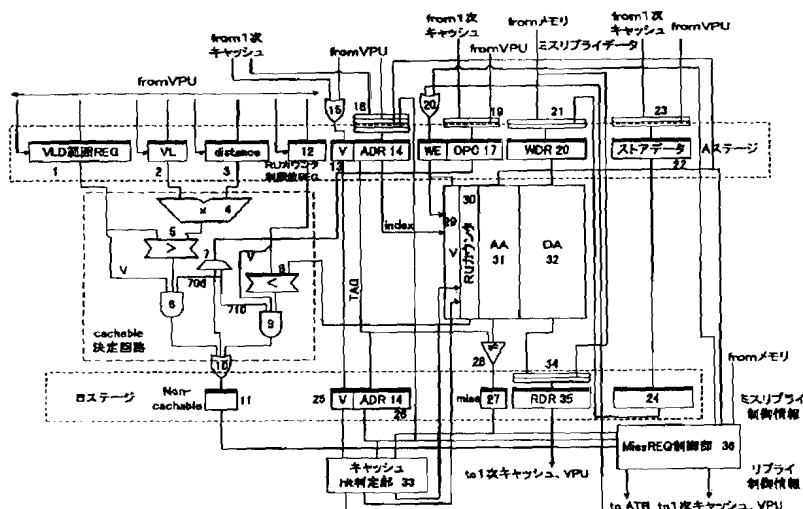
【図1】



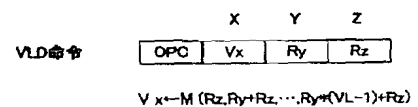
【図3】



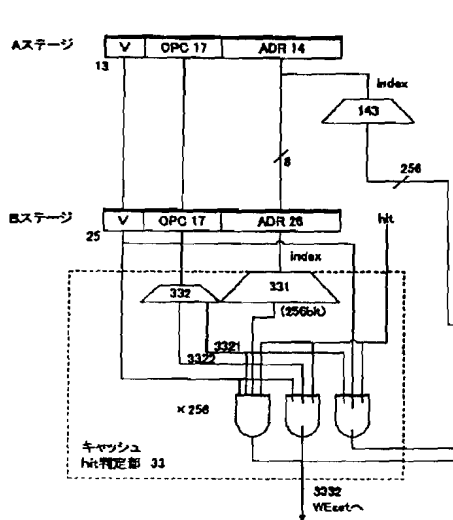
【図2】



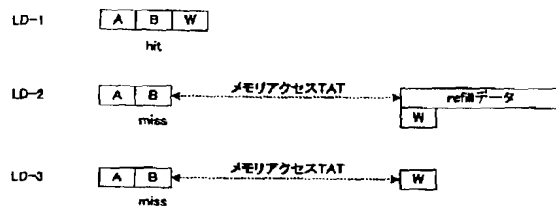
【図6】



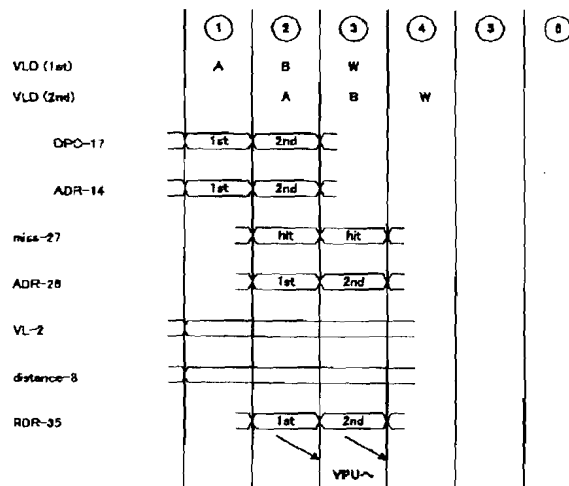
【図 4】



【図 5】

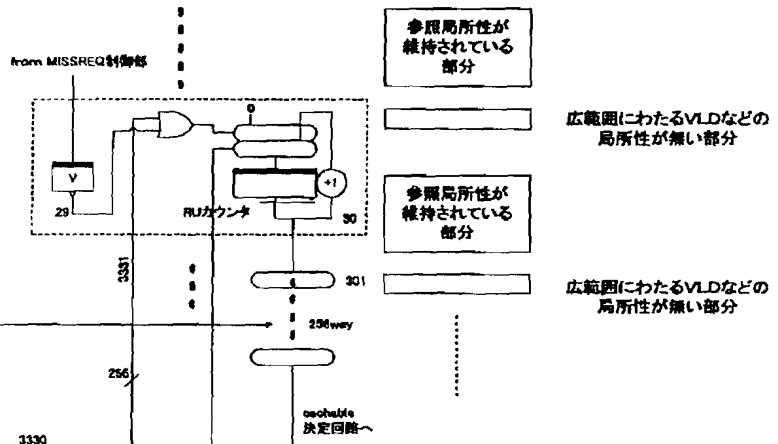


【図9】

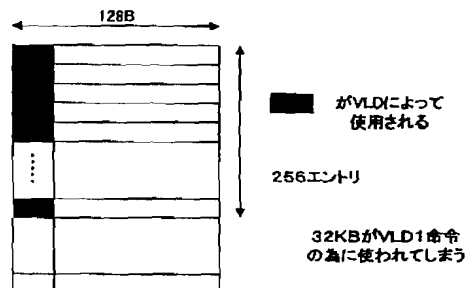


VL=2時のVLロ命令がキヤツシュした時の動作

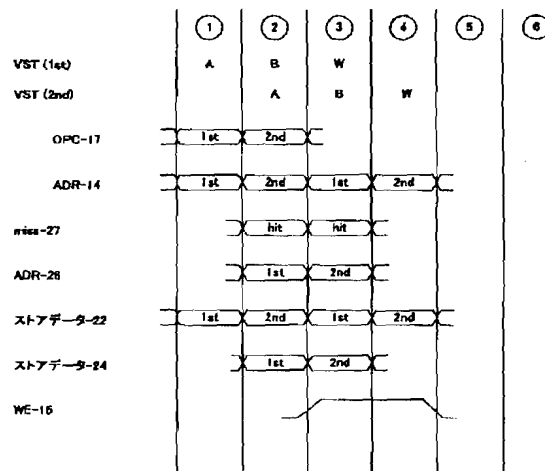
【図 7】



【图 8】

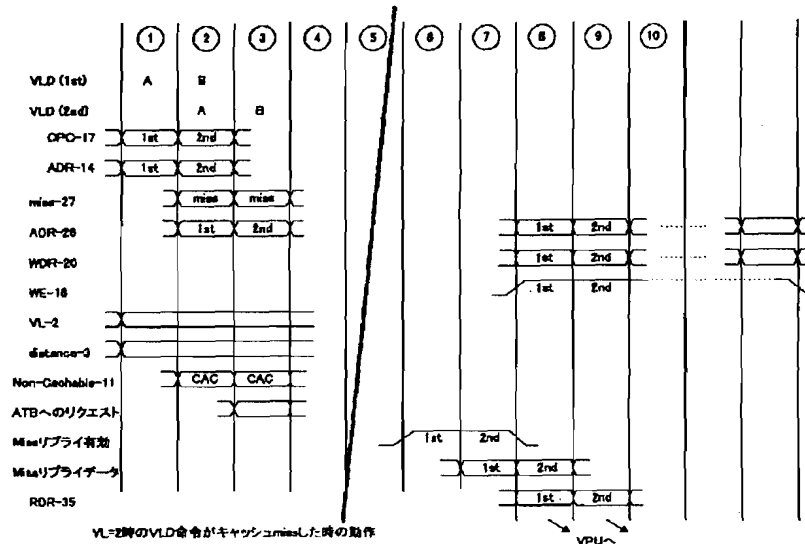


【図 10】

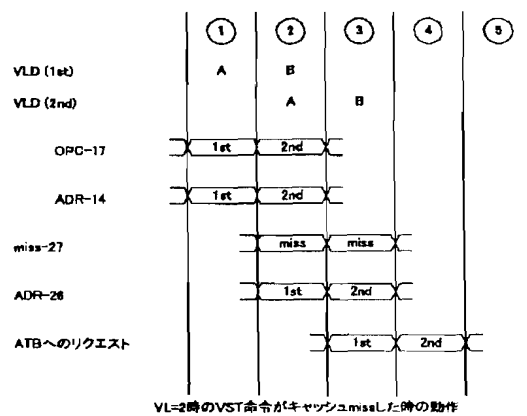


VL=2時のVST命令がキャッシュhitした時の動作

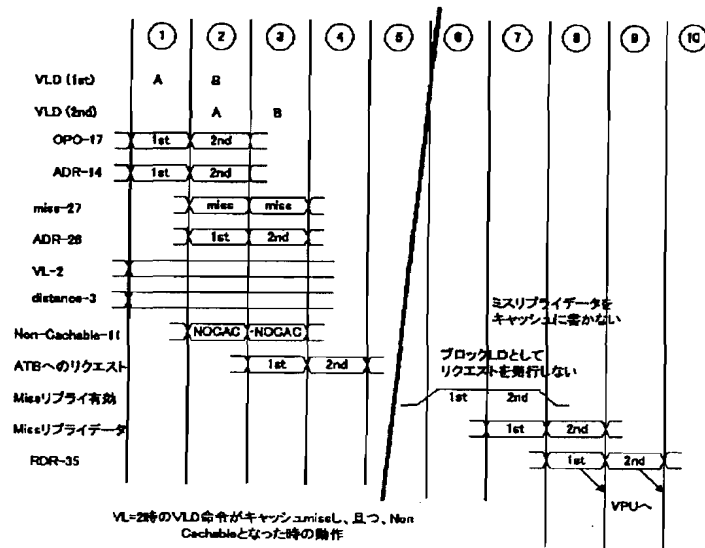
【図11】



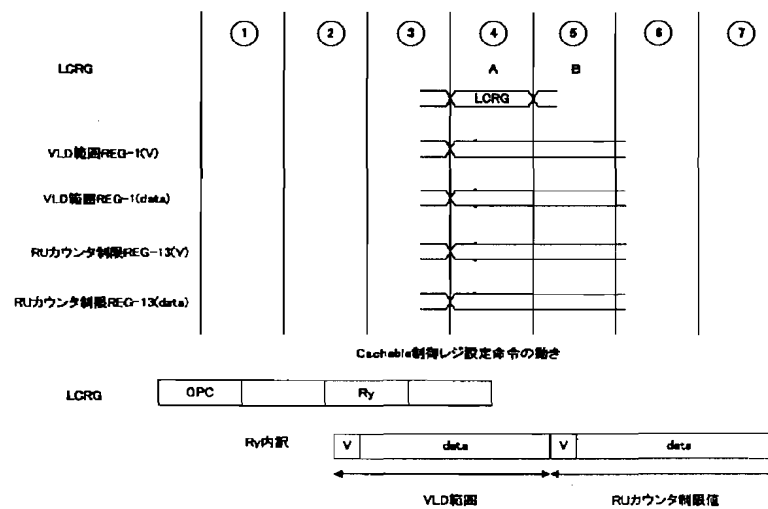
【図12】



【図13】



【図14】



【図15】

